



## Python

# If Statements and Functions



# Recap: If Statements

- If condition is true execute the code. If not true do not execute.
- Python if statements consist of:
  - If keyword
  - Condition
  - Colon
  - If clause (indented block of code)

```
permission = 'student'  
if permission == 'student':  
    print('You may join the class')
```

# Recap: else Statement

- When the condition is False an else statement is executed.
  - The else statement does not require a condition. But does need a colon.

```
permission = 'student':  
if permission == 'student':  
    print('You may join the class')  
else:  
    print('You may not join the class')
```

# elif statements (else if)

- When there are more than two possibilities an **elif** statement can be used.
- Python **elif** statements consist of:
  - elif keyword
  - Condition
  - Colon
  - If clause (indented block of code)

# elif statement example

```
permission = input('Who are you?')
if permission == 'student':
    print('You may join the class')
elif permission == 'teacher':
    print('You may join the class')
else:
    print('You are may not join the class')
```

- The elif statement will only be checked after the initial if condition is False.
- You may have unlimited elif statements.
- Once a condition is found to be true the elif conditions below it will not be checked (executed).

# Indentation

- Many programming languages use specific characters e.g. {} to distinguish between blocks of code.
- Python, however, uses indentation.



4 spaces is  
recommended  
Consistency is  
the key

# Task

- Open stream.py
- Correct any syntax errors in the code.
- Indent the code

# The Answer

```
stream = 'science'
if stream == 'commerce':
    print('Accounting today')
elif stream == 'design':
    print('Photoshop today')
elif stream == 'arts':
    print('Academic English today')
elif stream == 'science':
    print('Python today')
else:
    print('Sorry not sure what stream you are in')
```

Colons X 5

Indent x 5

Closing  
quotation  
mark



# Functions

- A function is a block of code that performs a particular task when called.
- Functions are declared and defined using the keyword **'def'**. A function consists of:
  - Function name, parenthesis and colon `def greet () :`
  - List of parameters (arguments ), which are passed to the function when called.

```
def greet (who) :
```

- Block of Python program that defines what the function does.

```
def greet (who) :  
    print('Hello %s' % who)
```

# Functions Explained

- The keyword `def`, (define) is followed by a single space then the name of the function. The name should reflect the action that the function performs. Parameters are contained in parentheses (), parameters are separated by commas. The defining of function must end with a colon `:`. The code inside the function must be indented.

```
def add(a,b):  
    total = a + b  
    statement = 'The sum of %d and %d is %d' %(a, b, total)  
    return statement
```

```
print(add(4,5))
```

- What is the output of this function?

The sum of 4 and 5 is 9

# Return Statement

- The return statement causes your function to exit and generate a value when called. The return statement is used when a function is required to output a value when called.

```
def squared(x):  
    return x * x
```

```
a = squared(4)  
print(a)
```

# Built-in Function – Input()

- What is the output of this function? ( 6 and 7 entered by user)

```
def add(a, b):  
    total = a + b  
    statement = '%d and %d is %d' %(a, b, total)  
    return statement
```

```
a = input('Please enter a number: ')  
b = input('Please enter another number: ')  
print(add(a, b))
```

Please enter a number:6

Please enter a number:7

```
print('%d and %d is %d' %(a,b,total))
```

TypeError: %d format: a number is required, not str

- What is the solution to this runtime error?

# Solution

```
def add(a, b):  
    total = a + b  
    statement = '%d and %d is %d' %(a, b, total)  
    return statement  
  
a = int(input('Please enter a number: '))  
b = int(input('Please enter another number: '))  
print(add(a, b))
```

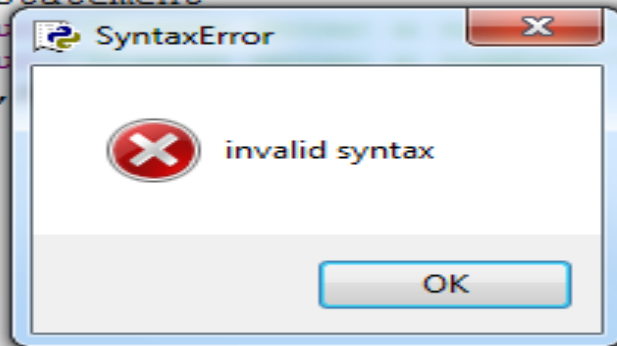
The int() method returns an integer.

When the user inputs 6 and 7 they are strings '6' '7'. The int() method converts these to integers (number).

# Errors

- What is the error in the following code:

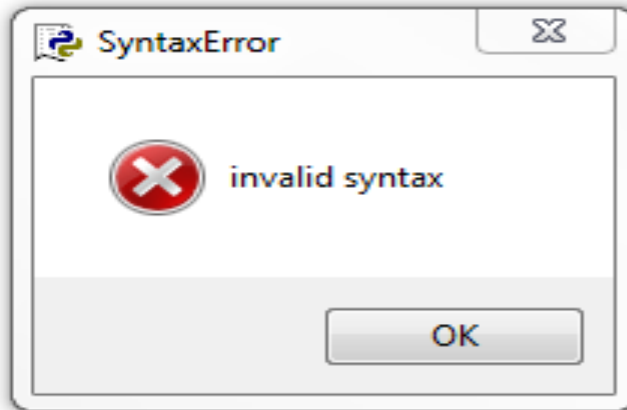
```
def add(a,b)
    total = a + b
    statement = '%d and %d is %d' % (a,b,total)
    return statement
a = int(input())
b = int(input())
print(add(a, b))
```



Computers are inflexible. Syntax is the way the code is written. If it is not perfect it is a syntax error.

# Computers make mistakes.

```
def add(a,b):  
    total = a + b  
    statement = '%d and %d is %d' %(a,b,total)  
    return statement  
a = int(input('Please enter a number:'))  
b = int(input('Please enter a number:'))  
print(add(a,b))
```



The mistake is:

```
a =int(input('Please enter a number:'))
```

However the computer has highlighted the next line.

# What is the error?

```
def add(a, b):  
    total = a + c  
    statement = '%d and %d is %d' %(a, b, total)  
    return statement
```

```
a = int(input('Please enter a number: '))  
b = int(input('Please enter another number: '))  
print(add(a, b))
```

add\_input.py", line 11, in add  
 total = a + c  
NameError: name 'c' is not defined



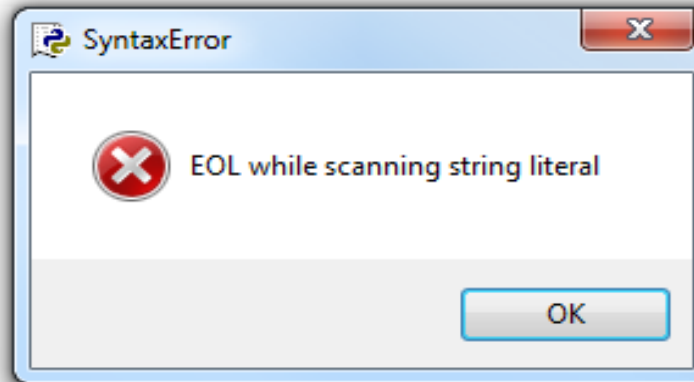
# What is the error?

```
def sum():
    a = int(input('Please enter a number:'))
    b = int(input('Please enter a number:'))
    total = a * b
    print('The multiplication of %s by %s is %s' % (a,b,total))
```

sum()

```
def sum():  
    a = int(input('Please enter a number:'))  
    b = int(input('Please enter a number:'))  
    total = a * b  
    print('The multiplication of %s by %s is %s' %(a,b,total))
```

sum()



## EOL – End of Line

**String Literal** – A set of characters enclosed in quotation marks (‘ ’)

# Local and Global Variables

- All variables defined in a function are local to the function. These are not recognized outside the function.
- Any variables defined outside the function but within the same module are global variables.

```
#Global variable
```

```
z = 10
```

```
def add_three(x,y):  
    return x + y + z
```

```
print(add_three(10,20))
```

40

```
def two_times(x,y)
```

```
#local variable
```

```
    z = x * y
```

```
    print('Inside the function z is %d ' %z)
```

```
    return x + y + z
```

```
print(two_times(10,20))
```

```
Print('z is still %d outside the function' %z)
```

Inside the function z is 200

230

z is still 10 outside the function

# Task

- Complete:
  - CAP Term 2S Lecture 2 - Python Quiz
  - CAP Term 2S Lecture 2 - Python Matching Functions Definitions



## Next Lecture

Python Loops and Lists

